professor ME. Pablo De Chiaro

Nivelamento — AEDII

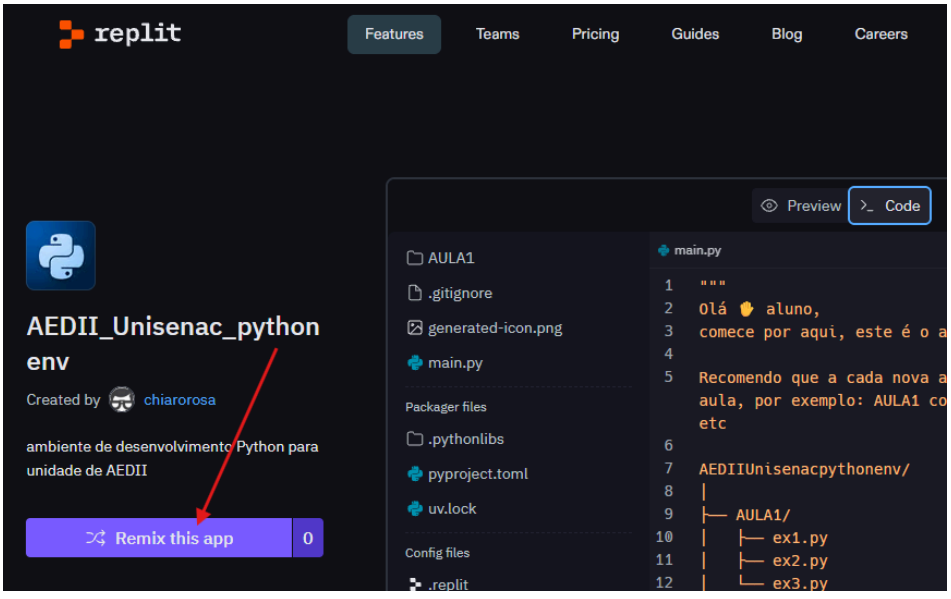
Data: Aula 1 - 05/08
Linguagens: Python ou JavaScript
Ambiente sugerido: Replit

Objetivos

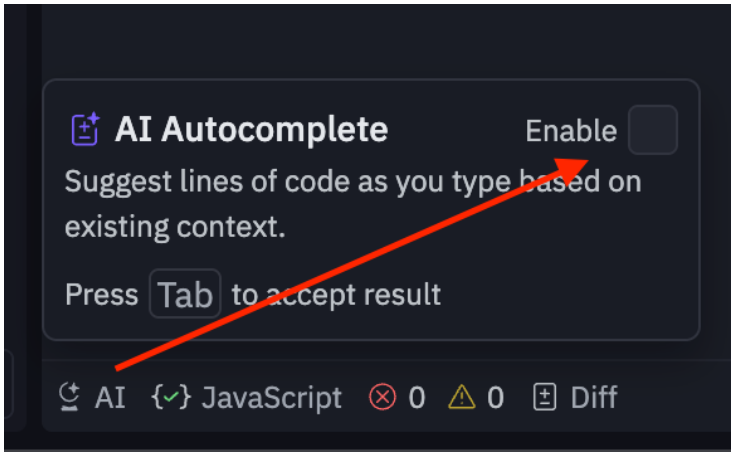
- Avaliar seu domínio dos conteúdos fundamentais de algoritmos e estruturas de dados.
- Incentivar autonomia, pesquisa e boa organização dos códigos.
- Desenvolver familiaridade com ambientes modernos de desenvolvimento e execução pelo terminal (CLI).

Instruções

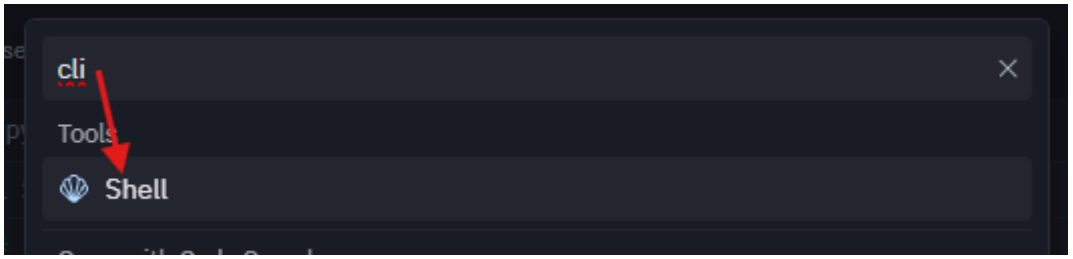
1. **Escolha uma das linguagens:** Python ou JavaScript (Node.js).
2. **Acesse o ambiente base:**
    - ambiente python
    - https://replit.com/@chiarorosa/AEDIIUnisenacpythonenv?v=1
    - ambiente JS (nodejs)
    - https://replit.com/@chiarorosa/AEDIIUnisenacjsenv?v=1
3. **Crie uma cópia para você** (Fork -Remix).



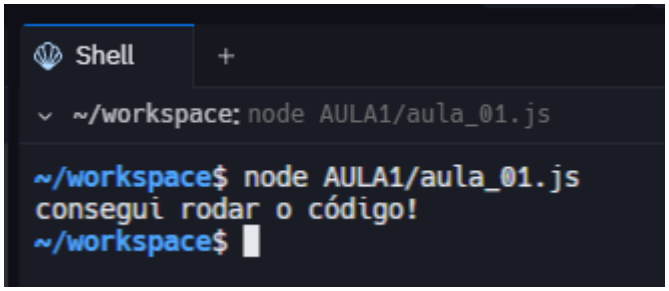
4. **Organize seu código da forma que preferir:**
   - **OBRIGATÓRIO** criar uma pasta por aula, ex: AULA1/.
   - Tudo em um arquivo único: aula_01.py ou aula_01.js **ou**
   - Arquivos separados: ex1.py, ex2.py... ou ex1.js, ex2.js…
   - **A AULA1 já foi criada no ambiente**
   - **OBRIGATÓRIO** desativar auxilio de IA no rodapé da interface



5. **Execute sempre pelo CLI**, as teclas CTRL+K pesquise CLI:

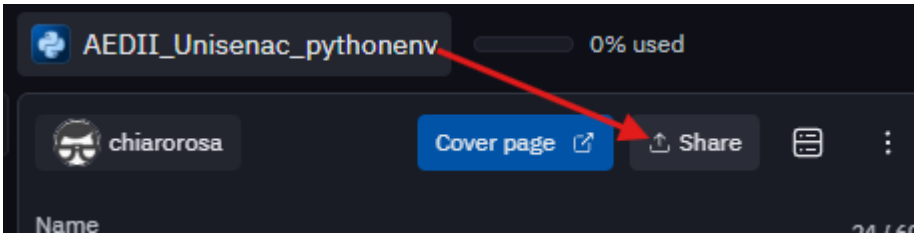


   - *python3 caminho/arquivo.py*
   - *node caminho/arquivo.js*
   - EXEMPLO **node AULA1/aula_01.js**
   - EXEMPLO **python3 AULA1/aula_01.py**



6. **Entrega:**
   - Clique no nome do seu projeto Replit, vá em **Share**, copie o link e envie conforme instruções do professor.

professor ME. Pablo De Chiaro



Materiais de Apoio

- Documentação Python
- Documentação JavaScript (MDN)

professor ME. Pablo De Chiaro

Lista de Exercícios

1 a 5 — Operadores, Strings e Estruturas Lineares

1. **Contar Dígitos Pares**
   Escreva um algoritmo que, dado um número inteiro, retorna a quantidade de dígitos pares existentes nele.

2. **Substituição de Caracteres em String**
   Implemente uma função que recebe uma string e retorna a mesma string com todas as vogais substituídas por '*'.

3. **Reversão com Pilha**
   Usando a estrutura de pilha, crie uma função que inverta uma palavra digitada pelo usuário.

4. **Simulação de Fila**
   Implemente uma fila simples para processar uma sequência de nomes (entrada e saída), mostrando a fila a cada operação.

5. **Remover Elementos Negativos de Lista**
   Dado um vetor/lista de inteiros, escreva um algoritmo que retorna uma nova lista apenas com os números não-negativos.

---

6 a 8 — Conjuntos, Dicionários e Algoritmos de Busca

6. **Remover Duplicatas Usando Conjunto**
   Implemente uma função que recebe uma lista de números e retorna uma nova lista sem elementos repetidos, usando conjuntos (set).

7. **Contagem de Palavras em String**
   Dada uma frase, escreva um algoritmo que conte quantas vezes cada palavra aparece, retornando um dicionário com o resultado.

8. **Busca em Dicionário de Contatos**
   Crie um dicionário com nome (chave) e telefone (valor). Implemente funções para buscar telefone pelo nome e listar os contatos em ordem alfabética.

---

professor ME. Pablo De Chiaro

9 e 10 — Recursividade

9. **Soma Recursiva de Vetor**
   Implemente uma função recursiva para somar todos os elementos de uma lista.

10. **Sequência de Fibonacci Recursiva**
    Escreva uma função recursiva que recebe um número n e retorna o n-ésimo termo da sequência de Fibonacci.

---

11 e 12 — Ordenação (Avançado)

11. **Implementar Bubble Sort**
    Implemente o algoritmo Bubble Sort para ordenar uma lista de números (sem usar métodos prontos).

12. **Ordenar Lista de Dicionários por Chave**
    Dada uma lista de dicionários com nomes e idades, implemente uma função que ordene essa lista pelo campo idade (ordem crescente).

Visão Geral do ambiente